Nejprve ke kapacitám a rezervám:

Pokud existuje jedno centrální téma všech algoritmů na toky, které je hlavní klíčovou myšlenkou, jsou to rezervy. Rezervy nám umožňují velmi jednodnuše vědět, kolik toku ještě můžeme hranou poslat, ale jejich hlavní a velmi-velmi důležitá funkce je, že nám umožňují předělávat již existující tok, a to zcela transparentně, aniž bychom museli explicitně uvažovat změny dříve postavených částí toku. Díky tomu tokové algoritmy v principu nemohou udělat špatnou volbu, která by jim zabránila získat tok, který je maximální. Algoritmy v6dy pracují s rezervami, nikdy nepracují přímo s kapacitami.
Rezerva r je definována pomocí kapacity c a aktuálního toku f:

$$r(vu) := c(vu) - f(vu) + f(uv)$$

Platí tedy $0 \leq r(vu) \leq c(vu) + c(uv)$, neboli pro kapacity ≤ 1 jsou rezervy ≤ 2.

Analýza 0/1 Dinicije

Úkol 3-1: Ukažte, že pokud původní graf měl kapacity pouze 1, blokující tok bude nalezen v čase $\mathcal{O}(m)$, a tedy i celková složitost algoritmu se příslušně změní (uveďte, jaká bude).

Řešení:

Uvažme jednu fázi hledání blokujícího toku. Ta probíhá na vrstevnaté síti. Hodnoty na hranách vrstevnaté sítě ("kapacity" vrstevnaté sítě) jsou rezervy původní sítě, a mají tedy hodnoty ≤ 2.
Blokující tok hledáme inkrementálně, hledáním zlepšujících cest a dočišťováním sítě. Čištění umíme odhadnout jako $\mathcal{O}(m)$, a konstrukce vrstevnaté sítě je $\mathcal{O}(n+m)$. Celková složitost fáze je tedy $\mathcal{O}(n+m)$. (Můžeme předpokládat souvislost výchozí sítě, čímž $\mathcal{O}(n+m) = \mathcal{O}(m)$. V opačném případě předzpracujeme vstup ještě před první fází prohledáváním v čase $\mathcal{O}(n+m)$, a zredukujeme vstup na souvsislý).
Nyní odhadneme složitost vyhledávání zlepšujících cest. Z vlastností vrstevnaté sítě víme, že kdykoliv algoritmus sáhne na hranu při vyhledávání, tato hrana bude součástí zlepšující cesty a její rezerva se sníží alespoň o 1. Z omezení na rezervy toto může nastat pro každou hranu nejvýše dvakrát. Celková složitost hledání je tedy $\mathcal{O}(m)$.
Použijeme-li odhad počtu fází $\mathcal{O}(n)$, celková složitost algoritmu je $\mathcal{O}(nm)$.
Alternativně (ačkoliv trochu zbytečně) můžeme počítat přes délky zlepšujících cest. Pokud v dané fázi je vzdálenost zdroje a stoku ve vrstevnaté síti d, každá zlepšující cesta bude mít délku d, bude nalezena v čase $\mathcal{O}(d)$, a sníží rezervy na d hranách. Protože snížení rezerv proběhne nejvýše $2m = \mathcal{O}(m)$, hledání cesty proběhne nejvýše $2m/d$-krát, s celkovou složitostí $\mathcal{O}(d\frac{m}{d}) = \mathcal{O}(m)$.
Ještě jinačí trik na dosažení příslušné složitosti bylo při konstrukci vrstevnaté sítě zdvojit všechny hrany s rezervou 2. Potom všechny hrany mají rezervu 1, a platí přímočarý argument, že každá zlepšující cesta nasytí (a tedy odstraní) všechny svoje hrany. Tento argument vlastně pouze přehazuje multiplikativní

konstantu z počtu projití jednou hranou před jejím smazáním na celkový počet hran. Hlavní nedostatek tohoto argumentu nicméně je, že upravuje chování algoritmu, a tedy není platný odhad složitosti Dinicova algoritmu.

Analýza 0/1 Goldberga

Úkol 3-2: Ukažte, že pokud původní graf měl kapacity pouze 1, umíme lépe odhadnout celkové zvýšení potenciálu, a tedy i celkový počet nenasycených převedení. Z toho přirozeně získáváme i lepší česovou složitost celého algoritmu (uveďte, jaká bude).
(Používáme stejný potenciál jako na přednášce, tedy součet výšek vrcholů s přebytkem. Stejně jako na přednášce, zpracováváme vrcholy s přebytkem v libovolném pořadí, nehledě na výšku - žádná vylepšení a heuristiky)

Řešení:

Nároky na přesnost v této úloze byly značně sníženy. Pojďme si tedy ukázat, jak vypadala obvyklá dobrá řešení.
Goldberův algoritmus (jako každý tokový algoritmus) pracuje s rezervami. Hodnoty na hranách jsou tedy ≤ 2.
Zkusme odhadnout počet nenasycených převedení přímo. Každé nenasycené převedení sníží rezervu hrany o 1. Tedy v konkrétní uvažované situaci dochází ke snížení z rezervy 2 na rezervu 1. Než může nastat další nenasycené převedení, musí rezerva hrany opět vzrůst, což nastane pouze převedením přebytku v opačném směru. Z fungování algoritmu se musí nejprve obrátit sklon hrany, tedy jeden z koncových vrcholů musí být zvednut (alespoň o 2).
Pro jednu konkrétní hranu tedy nemůžeme provést více nenasycených převedení, než kolik zvednutí proběhne na jejích koncových vrcholech. Protože takových zvednutí je $\mathcal{O}(n)$ (pro každou hranu), dostáváme, že celkový počet nenasycených převedení za celou dobu algoritmu je nejvýše $\mathcal{O}(nm)$.
Poznamenejme tady, že obvyklý pohled na situaci byl, že po nenasyceném převedení nutně následují nasycené, rezerva totiž klesla na 1, a další převedení ji vynuluje. Takové převedení ale nemusí nastat, z principu nenasyceného převedení byl vynulován přebytek v výchozím vrcholu hrany, můžeme tedy jako další akci algoritmu např. zvedat cílový vrchol hrany. Neexistuje tedy souvislost mezi počty jednotlivých typů převedení po směru dané hrany, pouze jejich horní odhady jsou stejné. Dá se ale použít trochu hlubší argument. Nenasycené převedení bylo možné pouze proto, že rezerva hrany byla 2, neboli rezerva hrany vedoucí v protisměru je 0. Aby bylo možné opět provést nenasycené převedení v tomto směru, musí se rezerva protější hrany opět vynulovat, to vyžaduje nasycené převedení v opačném směru. V tomto případě tedy existuje vztah mezi počtem nenasycených převedení po směru hrany, a nasycených převedení v protisměru. Všimněme si ale, že tento pohled nám neumožní vyřešit úlohu 3-3, protože pro vyšší kapacity není nutné rezervu v protisměru vynulovat, aby rezerva po směru byla ostře větší než 1.
Jaká je tedy celková složitost? Oba typy převedení zvládneme v konstantním čase, jejich celková složitost je tedy $\mathcal{O}(nm)$. Do hry ale vstupuje zvedání vrcholů,

protože zvednutí jednoho vrcholu nelze provést rychleji než v čase $\mathcal{O}(n)$, musíme totiž projít všechny sousedy (všechny incidentní hrany mohly změnit sklon). Zvedání lze tedy odhadnout jako $\mathcal{O}(n^3)$, což dává celkovou složitost $\mathcal{O}(n^3+nm)$. Můžeme ale odhadovat trochu lépe. Zvednutí vrcholu znamená projití všech jeho hran, zvednutím všech vrcholů tedy projdeme všechny hrany $\mathcal{O}(1)$-krát. Přesčítáme-li tedy složitosti zvedání přes jednotlivé hladiny, dostáváme celkovou složitost zvedání $\mathcal{O}(nm)$. Nyní můžeme konečně korektně prohlásit, že celková složitost algoritmu je $\mathcal{O}(nm)$.

Konstantní kapacity

Úkol 3-3: Miniúložka na závěr: Pokud bychom místo kapacit 1 dovolili (celočíselné) kapacity z rozmězí $1, \ldots, k$ pro nějakou konstantu $k = \mathcal{O}(1)$, co se změní v analýzách výše?

Řešení:

V obou případech projdou víceméně tytéž argumenty, pouze s jinými konstantami.
Pro Dinicův algoritmus platí, že při hledání blokujícího toku algoritmus sáhne na každou hranu (a tedy sníží její rezervu) maximálně $2k$-krát. Celková složitost hledání zlepšujících cest je tedy $\mathcal{O}(mk)$, což z předpokladu, že $k = \mathcal{O}(1)$, je rovno $\mathcal{O}(m)$. I celková složitost algoritmu tedy bude opět $\mathcal{O}(nm)$ (případně $\mathcal{O}(knm)$).
V případě Goldbergova algoritmu můžeme opět použít analýzu přes jednotlivé hrany. Pro každou hranu platí, že mezi dvěma změnami sklonu dojde k nejvýše $\mathcal{O}(k)$ nenasyceným převedením. Počet nenasycených převedení je tedy $\mathcal{O}(knm) = \mathcal{O}(nm)$ a celková složitost je opět $\mathcal{O}(nm)$ (případně $\mathcal{O}(knm)$).